GENIE User Manual:
Global biogeochemical cycling

Andy Ridgwell

October 3, 2008

Contents

1 README blurb **3**

1.1 Naming conventions blurb 3

1.2 Model background blurb 4

2 Getting your grubby little mitts on the model **6**

2.1 Where am I ... ? (what is in the file-tree?) 6

2.2 Configuring GENIE for you local software environment 6

3 Running GENIE **7**

3.1 ... manually ... :(. 7

3.2 ... via a shell script :) 8

3.3 Using `old_rungenie.sh` 8

3.4 Job submission to a cluster 10

4 Specifying time-dependent changes to tracer concentrations ('forcings') **12**

4.1 Tracer forcing as an explicit 2/3-D distributions 13

4.2 Tracer forcing as a point source or uniform distribution 14

5 Saving data **16**

5.1 'Time-slices' 16

5.2 'Time-series' 16

5.3 Results files 17

5.4 netCDF format output 18

5.4.1 Plotting with Panoply 18

5.4.2 MUTLAB plotting 18

5.5 Re-start files 19

A Quick-start guide **20**

B Available GENIE modules **22**

C GENIE directory structure **23**

D Namelist parameter definitions and defaults **28**

E The old_rungenie.sh script **29**
E.1 Overview . . . 29
E.2 technical details . . . 29

F FAQ (aka: has this dumb question been asked before?) **33**
F.1 Running experiments: General . . . 33
F.1.1 Help! My experiment has died 33
F.1.2 When do I have to recompile GENIE? . . . 33
F.2 Climate . . . 34
F.3 Biogeochemistry . . . 34
F.3.1 Can I separate out solubility (SST) related changes from stratification and circulation changes? 34
F.3.2 What is 'tracer auditing' – should I have it switched on? . . . 35
F.3.3 How do I do an ocean CO2 injection experiment? . . . 36
F.3.4 I have been told 'ERROR MESSAGE: Particulate tracer CaCO3 does does not have the corresponding
F.4 Running experiments: The almond.ggy.bris.ac.uk cluster . . . 37
F.4.1 Do I have to submit experiments to the queue rather than running interactively? . . . 37
F.4.2 Can I leave all my experiment results on the cluster for ever? . . . 37

G Known issues **38**
G.1 Radiocarbon tolerance . . . 38
G.2 Ocean tracer number related compilation problems . . . 38
G.3 Stack space . . . 38
G.4 Re-starts . . . 38

H Contact Information **40**

1 README blurb

This document covers the directory structure of the source code and ancillary data input files, instructions for compiling the GENIE executable, and how to configure and execute an experiment (i.e., how to get some actual work done). Also covered is how to specify the results you want saved and at what points in time, in what format they are saved, and (some ways of) getting the results visualized.

A Quick Start Guide is provided in Appendix A (page 20) if you prefer to just jump in semi-blind. However, it is advisable to at least read the FAQ section, lest any of the FAQs become EMFAQs (Even More Frequently Asked Questions) ...

For additional information refer to the companion **Tutorial** (`genie-tutorial`) and **HOW-TO** (`genie-howto`) documents. The source for this document is in TEX and resides on the SVN repository under:

`~/genie/genie-main/doc`

All the documentation (including this) can be generated from the TEX source in both postscript (.ps) and Adobe (.pdf) formats. To build documentation from the TEX source – from witin (`~/genie-main/doc`)[1], issue the command[2]:

- `$ make user-manual`

  Builds the oritinal GENIE user manual. [NOW REDUNDANT – SEE GENIE WIKI]
- `$ make genie-user-manual`

  Builds the (biogeochemistry and GENIE-1 focused) user manual.
- `$ make genie-tutorial`

  Builds documentation describing various tutorials and example model configurations and experimental designs.
- `$ make genie-howto`

  The HOW-TO documentation (potted explanations of how to get stuff done).

To clean up previously built documentation:

- `$ make clean`

  Cleans the built user-manual documentation files.
- `$ make clean-genie`

  Cleans all the built files for the genie-* documentation.

PLEASE edit and update the documentation and populate it with wonderful useful things. Spread the love :o)

1.1 Naming conventions blurb

The modular Earth system model is called GENIE, which stands for: Grid ENabled Integrated Earth system model. There are 2 primary variants - one with a simple (2-D, EMBM - see below) atmosphere referred to as 'GENIE-1', and one which employs a 3-D dynamical GCM atmosphere, called GENIE-2. In addition, there are (temporary) developmental variants that branch off of the primary model development pathway (the 'trunk') and which reside on 'branches' (all handled by the Subversion (SVN) code repository).

We will assume that the model directory structure (`genie/`), results directory, and (optional) model configuration directory and run scripts all reside in a directory given by `~`, which may look something like:

`home/username`

[1] ~ stands for the home directory
[2] At the `$` prompt

where *username* is your username. Model output (the directory where the results of the model experiments will be saved) will then appear here:

`home/username/genie_output`

or

`~/genie_output`

Rather than having multiple directories and ancillary files all cluttering up your home directory level and making it look untidy, the everything can be placed at an alternative location, e.g.:

`~/my_model_stuff/genie`

To install and run GENIE with a non-standard directory structure, the following files in `~/genie/genie-main` will have to be edited[3]:

```
user.mak
user.sh
```

The science modules currently available are listed in Appendix B (page 22).

1.2 Model background blurb

The core model component is the climate model C-GOLDSTEIN, as described in *Edwards and Marsh* [2005] (calibrated in *Hargreaves et al.* [2005] with minor bug fixes described in *Ridgwell et al.* [2007a]). It comprises three modules:

- GOLDSTEIN (the 3-D ocean circulation model of *Edwards and Shepherd* [2001])
- dynamic-thermodynamic sea-ice, and
- 2-D EMBM (Energy-Moisture Balance Model) following *Weaver et al.* [2001].

In GENIE-1 terminology, this EMBM atmosphere plus GOLDSTEIN ocean plus sea-ice (linked to GOLDSTEIN) has the 'flavor' (i.e., specific combination of science modules):

`genie_eb_go_gs`

The addition of ocean (and atmosphere) biogeochemistry gives you a coupled climate - carbon cycle model which was called CB-GOLDSTEIN in its non-modular, pre-GENIE incarnation. In addition to the GOLDSTEIN+sea-ice+EMBM flavor; `genie_eb_go_gs`, CB-GOLDSTEIN has an ocean biogeochemistry module called BIOGEM (for 'BIOGEochemistry Model'), a module to update and keep track of the atmospheric 'chemistry' (greenhouse gas concentration and isotopic composition) called ATCHEM ('ATmospheric CHEmistry Model'), and a coupler to exchange fluxes between ocean and atmosphere across the air-sea interface. The coupler also provides BIOGEM with information about the overlying atmospheric composition, and if required, can drive the EMBM with trace gas concentrations for calculating radiative forcing of climate. As well as indirect coupling via the atmosphere and EMBM, BIOGEM and GOLDSTEIN are directly coupled through the exchange of 3-D array information regarding the geochemical composition of the ocean. BIOGEM also makes copies of various key configuration parameters at start-up as well as paying attention to sea-ice extent etc, and provides these all as output (independently of GOLDSTEIN). The GENIE-1 flavor equivalent to CB-GOLDSTEIN is:

`genie_eb_go_gs_ac_bg`

The configuration, calibration, and evaluation of GENIE-1, is described in *Ridgwell et al.* [2007a]. Earlier incarnations of the ocean biogeochemistry are described in *Cameron et al.* [2005] and *Lenton et al.* [2006]. The global carbon cycle can be further extended with the addition of a representation of the mass balance of solid material deposited to deep-ocean sediments - a module called SEDGEM ('SEDiment GEochemistry Model'). This is coupled via an interfacing module to BIOGEM. The overall coupled climate-ocean biogeochemistry-sediment geochemistry model was previously termed CBS-GOLDSTEIN, but in GENIE-1 terminology is:

[3]When editing these files, particularly in Windoz, remember that these files must be executable.

genie_eb_go_gs_ac_bg_sg

The configuration, calibration, evaluation, and application of GENIE-1 with deep-sea sediments, is described in *Ridgwell and Hargreaves* [2007] and *Ridgwell et al.* [2007b]. Further developments to the sediment model are described in *Ridgwell* [2007].

When allowing burial of biogenic material in deep-sea sediments using the SEDGEM module, a further module is available to provide the (long-term) balance the loss of matter from the ocean. This is called ROKGEM (for: 'ROcK GEochemistry Model') and calculates the solute supply to the coastal ocean resulting from the weathering on land of exposed rock surfaces and soil minerals. The flavor of GENIE run with this additional weathering module is then termed:

genie_eb_go_gs_ac_bg_sg_rg

The Table 2 in Appendix B summarizes the different science modules in GENIE.

Suggested background readings to help understanding the model, its typical applications, its successes and perhaps more importantly its limitations are:

- *Annan et al.* [2005]
- *Edwards and Marsh* [2005] (for the fast climate model core component, C-GOLDSTEIN)
- *Edwards and Shepherd* [2001]
- *Hargreaves et al.* [2004]

For atmosphere-ocean(-sediment) carbon cycling and biogeochemistry, try:

- *Ridgwell et al.* [2007a,b]
- *Ridgwell and Hargreaves* [2007]
- *Ridgwell* [2007]
- *Cameron et al.* [2005] (for an earlier representation of biogeochemical cycling in the ocean and atmosphere)
- *Lenton et al.* [2006]
- *Ridgwell* [2001]
- *Ridgwell et al.* [2002]

Versions of most of the relevant model references can either be downloaded from:

- the 'official' GENIE project site, or
- my own publications server.

Also refer to the companion documentation to this manual:

- `genie-tutorial.tex` – Documentation describing various tutorials and example model configurations and experimental designs.
- `genie-howto.tex` –' HOW-TO' documentation (potted explanations of how to get stuff done).

2 Getting your grubby little mitts on the model

First, you will a copy of Subversion (abbreviated: 'SVN') installed on the machine you want to run the model on (or at least, on the machine you want the source code to be maintained on). See:

- the Wikipedia entry or
- the GENIE wiki

for a basic over-view of obtaining the source code (wiki username and password required ...). In summary:

If you are using linux, then you will most likely have the command line client already installed. Try typing:

```
$ which svn
```

to see if the client is in your path. Command line and GUI clients for SVN are available for just about any operating system. `tortoisesvn` is popular for windows. `SVNx` is a good choice for the Mac.

Once you have Subversion installed, you will be prompted for your password when required. SVN caches your password, and so you often do not need to type it.

You can get ('check out'[4]) a copy of GENIE from SVN by typing:

```
$ svn co http://source.ggy.bris.ac.uk/subversion/genie/trunk
--username=your_username ~/genie
```

To update an existing checkout, simply `cd` to the highest-level directory (`~`) and issue the update command:

```
$ svn update
```

note that you could update just a sub-tree of your checkout by changing directory to, say

```
~/genie/genie-embm
```

and then calling

```
$ svn update
```

which would result in an updating of just the EMBM module. Or you could just update a single file with `svn update` *`filename`* from within the appropriate directory, or by additionally supplying the path.

2.1 Where am I ... ? (what is in the file-tree?)

The pertinent directories associated with configuring and running experiments with the carbon cycle and (EMBM-based) climate using GENIE-1 are listed in full in Appendix C (page 23).

2.2 Configuring GENIE for you local software environment

Some hints for instally and adapting GENIE for your local software environment (and machine architecture) can be found in the documents `README` and `readme.main` in `~/genie-main/docs` but note that they are specific to GENIE (not genie) and refer to CVS rather than SVN for code repository access (as well as some other out-of-date details). no doubt these will be updated in due course ...

A general GENIE user-manual can also be created in a variety of formats (including .ps and .pdf) from within `~/genie/genie-main/doc` by issuing the command:

```
$ make user-manual
```

However, this currently suffers from some of the same documentation issues as the ASCII files in genie-main is as much as it is specific to GENIE (trunk) and assumes use of a CVS (rather than SVN) code repository. Again, an update will follow ... (ha ha ha ha)

[4] `svn co` is synonimous with `svn checkout`

3 Running GENIE

I'll discuss this in the reverse order just for the hell of it – setting a model experiment going *first* ... and only then describing what you should have done in the first place to have set it up correctly ;)

It is important to recognize from the outset that there are two fundamental modes of operating GENIE - as a new or continuing run. A new run is one which starts off from a default state of the system.

- In the case of the ocean model GOLDSTEIN, this is with the ocean set to uniform temperature and salinity throughout[5], and no momentum.
- In BIOGEM, the dissolved tracers are initialized with default values applied to all wet cells while all particulate concentrations in the ocean are zero.
- In ATCHEM all tracer gas concentrations (and isotopic properties) are initialized with uniform default values everywhere in the atmospheric grid.
- In SEDGEM the sediments are set to a homogeneous (detrital or ash tracer) composition throughout.

A continuing run will carry on from where-ever a previous run left off, using a set of files called 'restart' files that between them completely define a snapshot state of the system. More on this (plus examples) later (and in the tutorial document).

For GENIE-1, a model experiment can be initiated in two different ways ...

3.1 ... manually ... :(

The basic flavor and configuration of GENIE is run from `~/genie/genie-main` by typing:

```
./genie_example.job
```

`genie_example.job` is a shell script which sets up a basic ('base') configuration of the model. A different flavor and configuration of the model is enabled by specifying an alternative base configuration file:

```
./genie_example.job -f another.config
```

where `another.config` is a specified model configuration (/flavor) `.config file`.

GENIE is supplied with a number of pre-prepared *base config* files. These live in:

```
~/genie/genie-main/configs
```

The names of the *base config* files are typically structured as:

```
genie_aa_oo_ss_xxxx.config
```

where:

- `aa` is the atmosphere name (`ig`=igcm, `eb`=embm)
- `oo` is the ocean name (`fi`=fixed, `go`=goldstein ocean, `sl`=slab)
- `ss` is the sea-ice name (fi=fixed,gs=goldstein seaice, sl=slab), and
- xxxx is for any more info (e.g., `ac`=ATCHEM, `bg`=BIOGEM, `sg`=SEDGEM, `rg`=ROKGEM).

A relatively simple configuration file is `genie_ig_fi_fi.config` – a dynamical atmosphere (IGCM) plus prescribed ocean surface and sea-ice boundary conditions. In addition to specifying the flavor and grid resolution, this configuration file also changes the run-length from one month to 10 years and turns on daily IGCM atmosphere output. It also changes the name of the directory in which the experimental output is placed.

[5] See HOWTO for details of how to initialize with non-uniform initial T and S distributions.

3.2 ... via a shell script :)

For configuring and running experiments interactively or submitting jobs to a cluster queue, an executable shell script called `old_rungenie.sh`[6] is available [7]. `old_rungenie.sh` is invoked from ~ (i.e., the same level as the `genie` code tree and `genie_output` results directory).

If you are going to run the model using `old_rungenie.sh`, several additional directories are recommended which you must create:

- `~/genie_userconfigs` – Directory containing user configuration files[8]. These file contain namelist parameter values specific to a particular model experiment. These details are appended onto a basic configuration (`.config`) file (specifying general flavor and model configuration) – see Appendix E (page 29) for full details regarding what exactly `old_rungenie.sh` does.
- `~/genie_forcings` – Directory where prescribed tracer *forcings* are defined[9]. The location of this directory is defined in a namelist parameter (`bg_par_fordir_name`). The default is to use no tracer forcings, which is specified by the default forcing files found in the directory:
  `~/genie/genie-biogem/data/input}`
- `~/genie_archive` – Directory containing `tar`-ed and `gzip`-ed results archives.
- `~/genie_log` – Directory for storing experiment job logs.

3.3 Using `old_rungenie.sh`

You use `old_rungenie.sh`[10] to run GENIE, taking example flavor `genie_eb_go_gs_ac_bg` and experiment design (configuration patch file) `worbe2_preindustrial`) as follows:

```
$ ./old_rungenie.sh <options>
```

Now pay attention: there are 4 (FOUR) parameters that **must** be passed to the script as `<options>`:

1. The name of the required flavor and basic configuration, but omitting the '`.config`' bit, e.g.[11]:

   `genie_eb_go_gs_ac_bg`

2. The full path of the directory containing the user (experiment) configuration file (i.e., the file containing the list of namelist changes for a particular experiment). In the example, this might be something like:

   `~/genie_userconfigs/`

3. The name of the experiment, and specification of *user config* file is to be used, i.e.:

   `worbe2_preindustrial`

4. The run length of the experiment **in years**[12].

A typical command line to run the model might thus look something like (all one line and be careful with the S P A C E S):

```
./rungenie.sh genie_eb_go_gs_ac_bg
~/genie_userconfigs/ worbe2_preindustrial 10001
```

5. The full path (and name) of any restart required[13].

If the 5th (optional) parameter is not passed then GENIE will run from 'cold' (i.e., no re-start). If the 5th parameter is present then GENIE will attempt to run from a previously generated re-start state.

[6] The `old_` bit designates use of the non-xml configuration framework.
[7] This can be obtained from `mygenie.seao2.org`.
[8] This directory can be anywhere as it is defined in a passed parameter to `rungenie.sh` - see main body of text.
[9] Can be elsewhere - see main body of text.
[10] Note that the script file must be *executable* in order to run it: if not, set the correct permissions by entering the command at the linux command line: `chmod u+x old_rungenie.sh`
[11] Please note that *this is an example*, and does not necessarily mean that this is the same `.config` file as you want to use ...
[12] This must be entered on the command line as an integer, even though GENIE will be treating it as a real.
[13] There must exist a subdirectory in `~/genie_output` with this name.

A typical command line to run the model using a specified re-start might thus look something like (all one line):

```
./rungenie.sh genie_eb_go_gs_ac_bg
~/genie_userconfigs/ worbe2_historical 2001
~/genie_output/genie_eb_go_gs_ac_bg.worbe2_preindustrial
```

in which the experiment `worbe2_historical` will be run for 2001 years, using the flavor and basic configuration specified by `genie_eb_go_gs_ac_bg`, and using as a restart, the experiment saved in this location:

```
~/genie_output/genie_eb_go_gs_ac_bg.worbe2_preindustrial
```

In this methodology, for each model experiment you will require a plain text (ASCII) file that details any namelist parameter changes compared to the GENIE defaults and the basic flavor and configuration defined in the `.config` file. If you require any tracer forcings then you will also need to specify the directory containing the necessary forcing specification files . The location of the directory where GENIE will go looking for forcing specification files is set by a namelist parameter (`bg_par_fordir_name`). Full details of the specification and application of tracer forcings is given later.

Full details regarding what `old_rungenie.sh` actually does can be found in Appendix E (page 29).

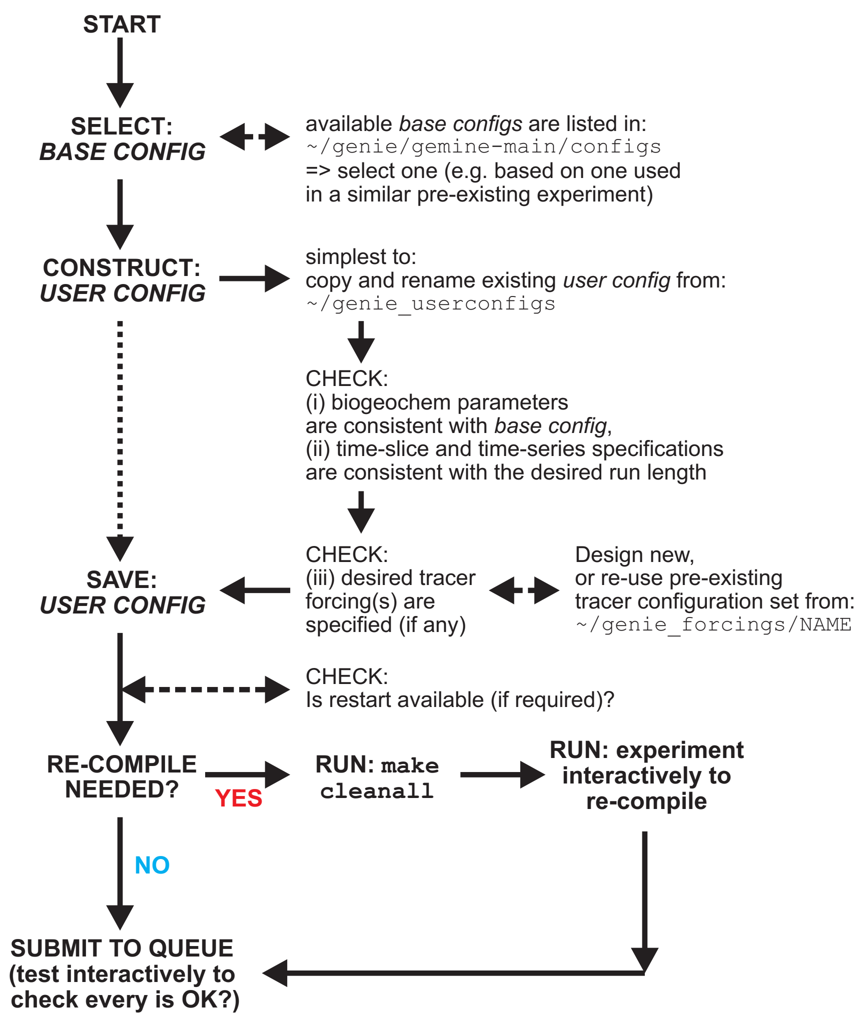

Figure 1: Schematic of the sequence-of-events in configuring and running an experiment using the `old_rungenie.sh` methodology.

3.4 Job submission to a cluster

For submitting jobs with Sun Grid Engine (SGE) on a cluster:[14]:

```
$ qsub -S /bin/bash subgenie.sh <options>
```

Here: the `-S /bin/bash` part is to ensure that SGE uses the BASH shell to submit the job because this is the language that `subgenie.sh` has been written in.

Take care that the installed FORTRAN compiler can be seen by the cluster nodes. If not, the GENIE executable must have already been built prior to submitting a job. The easiest way to do this is to run GENIE interactively briefly (e.g., with a run length of just a couple of years or kill it) and then submit the full run to the cluster.

Other useful submission options for SGE:

- To redirect the standard output stream:
  ```
  $ qsub -o genie_log ...
  ```
- To redirect the standard error stream:
  ```
  $ qsub -e genie_log ...
  ```
- To merge standard output and error streams into standard output:
  ```
  $ qsub -j y ...
  ```
- To specify particular resources, such as the nodes with 8 GB of RAM:
  ```
  $ qsub -l mem_total = 8.0G ...
  ```
- To decrease the priority of a job[15]:
  ```
  $ qsub -p 1 ...
  ```
- To submit a job from the current working directory:
  ```
  $ qsub cwd
  ```
- Request an email is sent when the job starts and/or when it finishes – see the man pages for `qsub` for the required syntax).
- A complete example would be:
  ```
  $ qsub -j y -o genie_log -S /bin/bash old_rungenie.sh <options>
  ```
  which merges standard output and error streams and redirects the resulting file to the directory ~/genie_log. Note that to redirect output as per in this particular example, the directory `~genie_log` **MUST** be present. I have no idea what happens if it is not ... but it can't be good ;)

You can check the status of the SGE job queue with the command:

```
$ qstat -f
```

and you can kill a job with the `qdel` command, the job numbers being given by the `qstat` command.

On `almond.ggy.bris.ac.uk` you can graphically check what is going on via the www. In an X window, enter:

```
$ ssh almond.ggy.bris.ac.uk
$ firefox --no-remote &
```

and follow the links: 'Cluster Status (Ganglia)', and then 'Job Queue' to get to the actual queue listing and cluster status.

NOTE: It is common on clusters for the FORTRAN compiler to not be accessible by the computer nodes. `almond.ggy.bris.ac.uk` is no exception in this. The implication of this is that **the GENIE-1 modules must be already compiled BEFORE a job is submitted to the queue**.

In other words; if you have just issued a `make cleanall`, you MUST briefly run your desired experiment (or equivalent) interactively (i.e., in the shell window) to ensure that everything is correctly compiled.

[14] With a different queue management environment it may be necessary to place the call to `rungenie.sh` together with its list of parameters into an executable shell, and submit that instead.

[15] The default priority is 0. A lower priority has a higher value ... !

For instance, either run the experiment for 1 or 11 years say, or (better): start the experiment for the desired full duration, but 'kill it' (Ctrl-C) once the experiment is running successfully.

Remember that if you have just changed model resolution or number of tracers you must force a complete recompile by issuing a `make cleanall` (and then start the model running interactively in a shell).

4 Specifying time-dependent changes to tracer concentrations ('forcings')

Even the most highly coupled instance of GENIE is far from being a complete description of the Earth system. For instance, the lack of a dynamic atmosphere able to perform tracer advection means that dust deposition to the ocean surface cannot be internally calculated. With respect to longer-term biogeochemical cycling, the model currently lacks any explicit representation of shallow water carbonate deposition in banks and reefs. In terms of global change and anthropogenic impacts on carbon cycling and climate, GENIE is also unable to predict fossil CO2 emissions from human activities (just how much do you want to burn in the future?). Provision is therefore made for prescribing flux fields or boundary conditions to the model. These are termed forcings.

There is a serious difficulty in devising a means for implementing a boundary condition to the model because the most interesting and relevant forcings tend to be time-dependent (i.e., the magnitudes vary with time). Worse, the boundary conditions are often spatially heterogeneous (i.e., the magnitude of the forcing is not the same at each and every grid point or cell). Dust fluxes are a good example of spatial heterogeneity in a flux input to the system. To fully define a time-varying 2D flux forcing for say 1000 years of model integration at a 0.01 yr time-step equates to $36 \times 36 \times 1000 \times 100$ data points – clearly beyond simple and efficient data manipulation and processing. One end-member solution would be to define the complete spatial distribution, but leave the fluxes invariant with respect to time. The opposite end-member would by to fully define the time-varying nature of the forcing but distribute the flux uniformly in space. Both approaches have serious drawbacks for many applications. A compromise scheme has therefore been devised to encapsulate data input and output as efficiently and as flexibly as possible.

Firstly, the forcing of atmosphere, ocean, and sediment tracers are treated separately. For the atmosphere (`atm_*`) tracers, atmospheric composition is manipulated from the 2-D surface ocean grid of BIOGEM. This means that no forcing is applied to the atmosphere overlying 'dry' land cells. For the ocean (`ocn_*`) tracers, all the 'wet' cells in of the full 3-D ocean grid of BIOGEM can be forced. For the sediment (`sed_*`) tracers, all the cells comprising the 3-D ocean grid of biogem (rather than the 2D SEDGEM grid) are manipulated[16].

Secondly, two different types of forcings of the system are recognized: a flux forcing and a restoring forcing. The first type, the flux forcing, is pretty self-evident - it represents a flux of tracer that is applied to each cell. The restoring forcing is less intuitive – a flux is applied to the grid cells with a value calculated to bring the tracer values closer to the prescribed restoring value. A time-scale (in years) determines the rate at which the tracer values are brought toward the boundary condition. This time-scale value is termed the restoring constant. The smaller the value of the restoring constant the 'harder' the restoring, and the more rapidly the model will be constrained to approach the boundary condition.

Whether a flux or restoring forcing is required together with the details of the forcing, and what tracer(s) is (/are) to be forced, are specified in a set of 3 tracer forcing configuration files; one for each type of tracer:

- `configure_forcings_atm.dat`
- `configure_forcings_ocn.dat`
- `configure_forcings_sed.dat`

Each file consists of rows of settings; one row for each different tracer. The settings are arranged into columns; the description of each column and allowed values are summarized at the bottom of each file. Options include whether a flux or restoring forcing is required, the time-constant of any restoring forcing, etc.

A set of tracer forcing configuration files with default values (for no forcings of any sort) live in:

[16]No provision is currently made for restoring forcing.

~/genie-biogem/data/input

Files to specify alternative forcing configurations will typically be located in sub-directories of (genie_forcings).

For a tracer forcing top be fully defined, one further set of files are required (and also located in sub-directories of (genie_forcings), taking filenames of the format:

- biogem_force_*_sig.dat

and contains two columns of information: the first is a time marker (year) and is paired with a corresponding tracer scalar value in the second column. These files all live happily together in the directory defined by the namelist parameter: (i,j). The information in the signal file define the time-varying information. For a model year falling outside of the maximum or minimum time markers specified in the signal file, no forcing will be applied (this in effect allows a restoring forcing to be turned 'on' and then 'off' again later). For model years inbetween specified time points, the tracer scalar values are linearly interpolation. It should be noted that biogem_force_*_sig.dat file must contain at least one (row) entry.

The units for the flux forcing for all tracers is mol yr-1 for bulk fluxes, and per mil for isotopic composition. For restoring forcings, the units are atmospheres (atm) for atmospheric (gaseous) tracers, and mol kg-1 for ocean (dissolved) tracers. For isotopic tracers, the units are per mil ().

There are then two alternative ways of providing the final information needed to fully describe a tracer forcing – the spatial component – by explicitly defining 2- or 3-D distributions (the sole, original methodology), or by assuming a much simpler, point source, or uniform distribution. The latter is rather easier to grasp and implement, but lack the flexibility of the former.

4.1 Tracer forcing as an explicit 2/3-D distributions

The tracer forcing in the explicit 2/3-D methodology requires two further files, with filenames:

- biogem_force_*_I.dat
- biogem_force_*_II.dat

These hold information about spatial distributions. The data format of these is either 2-D for the atmosphere or surface ocean only - rows (j) and columns (i), or 3-D (for the whole ocean) with the successive depth layers (k) repeated as (i,j) blocks down through the file.

The way in which these three files are employed to define a spatially-explicit time-varying input field is as follows. The first field file (*_I.dat) defines the baseline distribution and the second (*_II.dat) an alternative distribution. The difference between the two fields defines the spatial component of a time-varying forcing. The tracer scalar value in the signal file modifies the difference between the two spatial fields.

Putting it another way – the magnitude of the forcing that is applied at any location (i,j) and at point in time (t), $F_{(i,j),(t)}$, is equal to the baseline field, defined in $biogem\_force\_*\_I_{(i,j)}$, plus the time-dependent modifier ($S_{(t)}$) times the difference between the two fields, i.e.:

$$F_{(i,j),(t)} = biogem\_force\_*\_I_{(i,j)} + S(t) \times \left(biogem\_force\_*\_II_{(i,j)} - biogem\_force\_*\_I_{(i,j)}\right)$$

Remember that the value of the time-dependent tracer scalar (S(t)) is interpolated from the contents of the forcing signal file biogem_force_*_sig.dat.

Because what I have just written above is probably vanishingly indistinguishable from complete gobbledygook, a couple of examples might help (or not):

1. One way of using the forcing functionality of BIOGEM would be to assign a minimum forcing field to I, a maximum forcing field to II, and specify a normalized (i.e., taking values between 0.0 and 1.0) series of modifier values in the the signal file. For example, one could continually vary the surface temperature field seen by the marine carbon cycle over the deglacial transition, based on available end-member reconstructions for glacial maximum and modern [*CLIMAP*, 1980]) and by assuming a semi-representative (normalized) signal with which to interpolate between these two time-slice reconstructions (see *Ridgwell* [2001]).
2. A second way of using the functionality of tracer forcing in BIOGEM would be to set every wet cell in

biogem_force_*_I.dat to a value of 0.0, and every corresponding location in biogem_force_*_II.dat to a value of 1.0. The forcing applied to the model will then be the same as applying the value of the interpolated tracer scalar (in the signal file) equally to each and every cell, i.e.:
$F_{(i,j),(t)} = 0 + S_(t) \times \big(biogem\_force\_*\_II_{(i,j)} - 0\big)$
An example of this usage would be in applying a spatially uniform time-varying change in atmospheric composition, such as restoring atmospheric CO2 to an observed historical or predicted future atmospheric concentrations trajectory.

It should be noted that although this scheme is generic and can equally be applied to any tracer (atm, ocn, or sed and including isotopic properties) as well as offering reasonable flexibility in representing the time-varying characteristics of a boundary condition, it does have limitations and cannot cover all possible eventualities. For instance, historical changes in atmospheric CFC concentrations and CO2 radiocarbon activity vary not only with time, but the spatial heterogeneity changes in a complex way that cannot be represented as a interpolation between two alternative end-member distributions. In this way the built-in framework for tracer forcings in GENIE does not conform to the OCMIP protocol [REF] for ocean (carbon cycle) model evaluation.

4.2 Tracer forcing as a point source or uniform distribution

While the spatial forcing fields (biogem_force_*_I.dat, biogem_force_*_I.dat) provide maximum flexibility, they are not always so easy to use, nor perhaps, understand(!) ;) An alternative, simply way of specifying forcings has therefore been implemented. You can access this by adding the namelist:

```
bg_ctrl_force_oldformat=.false.
```

which simply tells BIOGEM to expect a new file format for the tracer forcing configuration files.

As before, the tracer forcing configuration files live in the directory specified by the namelist parameter bg_par_fordir_name e.g.:

```
bg_par_fordir_name="~/genie_forcings/worbe2_historical"
```

There are 3 files - one for each type of tracer (i.e., gas, dissolved, solid/particulate):

- configure_forcings_atm.dat – gases
- configure_forcings_ocn.dat – dissolved substances
- configure_forcings_sed.dat – solid substances / particulates

For instance, the format of the tracer forcing configuration file configure_forcings_atm.dat looks like:

```
 01 02    03 04 05 06  07  08
 \/ \/    \/ \/ \/ \/  \/  \/

-START-OF-DATA-
 F  0.0  F  F  F   2  01  01  '[surface air temperature (K)]'
 F  0.0  F  F  F   2  01  01  '[specific humidity (n/a)]'
 f  0.1  f  t  F   2  01  01  '[carbon dioxide (CO2) partial pressure (atm)]'
 f  0.1  f  t  F   2  01  01  '[d13C CO2 (o/oo)]'
 F  1.0  F  F  f   2  01  01  '[oxygen (O2) partial pressure (atm)]'
 F  1.0  F  F  F   2  01  01  '[d18O O2 (o/oo)]'
 F  1.0  F  F  F   2  01  01  '[nitrogen (N2) partial pressure (atm)]'
 F  1.0  F  F  F   2  01  01  '[d15N N2 (o/oo)]'
 F  1.0  F  F  F   2  01  01  '[methane (CH4) partial pressure (atm)]'
 F  1.0  F  F  F   2  01  01  '[d13C CH4 (o/oo)]'
 F  1.0  F  F  F   2  01  01  '[d14C CH4 (o/oo)]'
 F  1.0  F  F  F   2  01  01  '[sulphur hexafloride (SF6) partial pressure (atm)]'
 F  1.0  F  F  F   2  01  01  '[nitrous oxide (N2O) partial pressure (atm)]'
```

```
 F  1.0  F  F  F   2  01  01  '[d15N N2O (o/oo)]'
 F  1.0  F  F  F   2  01  01  '[hydrogen sulphide (H2S) partial pressure (atm)]'
 F  1.0  F  F  F   2  01  01  '[d32S of H2S (o/oo)]'
 f  0.1  F  F  F   2  01  01  '[CFC-11 partial pressure (atm)]'
 f  0.1  F  F  F   2  01  01  '[CFC-12 partial pressure (atm)]'
-END-OF-DATA-

 /\ /\ /\   /\ /\ /\  /\  /\  /\
 01 02 03   04 05 06  07  08  09
```

The description of each column is detailed at the bottom of the file:

```
COLUMN #01: include restoring forcing of tracer? [DATA FORMAT: 'T'/'F' ('t'/'f')]
COLUMN #02: time constant of restoring forcing (years) [DATA FORMAT: real]
COLUMN #03: include flux forcing of tracer? [DATA FORMAT: 'T'/'F' ('t'/'f')]
COLUMN #04: scale flux forcing of tracer? [DATA FORMAT: 'T'/'F' ('t'/'f')]
COLUMN #05: force ocean surface in equilibrium with atmosphere?
            (needs ocean restoring forcing set) [DATA FORMAT: 'T'/'F' ('t'/'f')]
COLUMN #06: make forcing uniform over this dimension
            (2 = 2D, 0 = point, ELSE spatially explicit forcing) [DATA FORMAT: integer]
COLUMN #07: i grid location of point forcing [DATA FORMAT: integer]
COLUMN #08: j grid location of point forcing [DATA FORMAT: integer]
COLUMN END: TRACER DESCRIPTION
```

The columns pertinent to the this alternative way of implementing tracer forcings are:

- `#01` - do you want to restore atmospheric composition toward a specified concentration (or isotopic ratio)? '`t`' for true ('yes'); '`f`' for false ('no'). Note that it does not matter whether you use upper-case (capitals) ('T') or lower-case ('`t`'). If selected, column `#2` determines the time-constant of the restoring.
- `#03` - (if not `#1`) do you want to apply a flux of gas to the atmosphere?
- `#06` - what degree of homogenization of the flux do you wish to apply? '`2`' == apply flux evenly throughout the atmosphere; '`0`' apply the flux at a single point only; ('`-1`' == use the original (more complex) flux forcing methodology with the flux or fractional weighting of total flux specified explicitly at each individual grid point). If 0, a forcing-at-a-point, the next 2 columns (`#7` and `#8`) then define the (`i,j`) location on the model grid (longitude ('`i`') and latitude ('`j`')) of the point source.

The forcing configuration files for dissolved tracers (`configure_forcings_ocn.dat`) and particulates (`configure_forcings_sed.dat`) should be self-explanatory; again there is a description of each column at the bottom of the file. However, for dissolved tracer forcings, there is an additional column for the '`k`' (depth level in the ocean) location of a flux applied at a single point, in addition to longitude ('`i`') and latitude ('`j`').

5 Saving data

This section covers how to generate results files from the model runs, and what file is which and which contains what. Confused? you soon will be ...

The following sections cover:

1. The two basic ways of saving data – as spatially explicit time-slices and as a continuous time-series of a single global (or otherwise representative) variable.
2. File naming conventions of results files .
3. Output produced in netCDF format.
4. Plotting using the MATLAB scripts in `genie-matlab`.
5. Restart files.

Results files as well as re-start files are all saved in module-specific results directories. So, for BIOGEM results files you will need to go to `~/genie_output/$RUNID/biogem`, where `$RUNID` is the model experiment name, and for SEDGEM results: `~/genie_output/$RUNID/sedgem`. Note that ATCHEM currently does not save its own data files (which are instead saved by BIGOEM by accessing the atmospheric composition information placed on the surface-atmosphere tracer array). Also note that SEDGEM saves data only at the end of an experiment[17], and does not have access to the time-series and time-slice information available to BIOGEM (or even the current 'time' during an experiment).

5.1 'Time-slices'

One of the most informative data sets that can be saved is that of the spatial distribution of properties. However, saving full spatial distributions (a $36 \times 36 \times 8$ array) for any or all of the tracers at each and every time-step is clearly not practical, not only in terms of data storage but also because of the detrimental effect that repeated disk access has on model run-time.

Instead, BIOGEM saves the full spatial distribution of tracer properties only at one or more predefined time points (years). These are called 'time-slices'. At the specified time points, t_n, a set of spatially-explicit data fields are saved for all the key tracer, flux, and physical characteristics of the system. Rather than taking an instantaneous snapshot, the time-slice is averaged over a specified integration interval Δt (in years), defined by the namelist parameter `bg_par_data_save_slice_dt`[18]. The model state is thus integrated from time $t_n - \Delta t/2$ to $t_n + \Delta t/2$. For instance, setting a value of $\Delta t = 1.0$ year results in all seasonal variability being removed from the saved time-slices, and successive time-slices then only reflect long-term (>1 year) trends in system state.

The mid-point years (t_n) for which time-slices should be saved are specified in a single column pain text (ASCII) file, whose name is specified by the namelist parameter `bg_par_infile_slice_name`[19]. The order in which the time sequence is ordered (i.e., ascending or descending time values) does not actually matter in practice as long as the list of times is ordered sequentially – the list will be internally re-ordered if necessary according to the selection of BP (the model running backwards-in-time) or not according to the logical value of the namelist parameter `bg_ctrl_misc_t_BP`, which is `.false.` by default.

5.2 'Time-series'

The second data format for model output is that of a time-series of of change in a single (integrated) property of the Earth system. Model characteristics must be reducible to a single meaningful variable for

[17] With the exception of sediment core location environmental properties, which is saved whenever the sediment column grows or contracts (by a 1 cm unit).

[18] An empty list is valid - time-slices will then be populated for you at an interval set by the time-slice integration interval. But if you really don't want any time-slices, just set the first (or only) time point to occur beyond the end year of the run.

[19] The location of this file is specified by the BIOGEM data input directory namelist parameter: `bg_par_indir_name` which by default is `~/genie-biogem/data/input`.

this to be practical (i.e., saving the time-varying nature of 3-D ocean tracer distributions is not). Suitable reduced indicators include: the total inventories in the ocean and/or atmosphere of various tracers (or equivalently, the mean global concentrations / partial pressures, respectively), global sea-ice coverage.

A file containing a series of model times (t_n) at which data points need be saved is defined as before in a file whose name is specified by the namelist parameter `bg_par_infile_sig_name`. The data values saved in the time-series file do not represent discrete values in time but an average, calculated from time $t_n - \Delta t/2$ to $t_n + \Delta t/2$ as per the construction of time-slices. The averaging interval, Δt, is set by the value of the namelist parameter `bg_par_data_save_sig_dt`. If there are less than two elements present in the list, a default frequency of data saving will be invoked, set equal the averaging interval, except in the situation that this results in an unreasonably large amount of data, when an order of magnitude (or more than one order of magnitude where necessary) fewer save points are assumed.[20]

5.3 Results files

At the end of an experiment[21] the BIOGEM results directory will contain some of all of the following files:

1. `fields_biogem_2d.nc` – 2-D fields of (mostly) ocean bottom, ocean surface, and sediment surface properties.[22] Also: water-column integrals of certain geochemistry diagnostics, air-sea gas exchange fluxes, atmospheric composition.
2. `fields_biogem_3d.nc` – 3-D fields ocean dissolved and particulate tracer properties.[23]
3. `biogem_series_*` – Time-series results file.
4. `biogem_year_*_diag_GLOBAL.res` – Miscellaneous global diagnostic information. It is saved at each requested time-slice with the file-name string containing the mid-point of the time-slice (as years). The diagnostics includes:
   - time mid-point and integration interval
   - global ocean surface area and volume
   - mean global sir-sea gas exchange coefficient (for CO_2)
   - mean atmospheric tracer concentrations plus total inventory
   - mean ocean tracer concentrations plus total inventory
   - mean plus total global productivity
   - mean plus total global sedimentation

In the SEDGEM results directory, some or all of the following files will be written:

1. `fields_sedgem_2d.nc` – Contains 2-D fields of sediment surface and ocean bottom properties.[24]
2. `sedcore_*` These files contain the stacked records of accumulated deep-sea sediment composition. Files with a name of the form sedcore_xxyy.res are also only saved at the very end of the model run. Each is a plain text (ASCII) file format containing the down-core properties of a specific sediment grid point. The `xxyy` identifier corresponds to the `(i,j)` grid location of the artificial sediment core as a zero-padded 4-digit integer. The data is saved in a space-delineated column format with column headers. In some of the column headers there is no space to include the units; for the solid components (e.g., POC, CaCO3, det, FeO, opal, ash) the units are either wt% (the default) or vol% depending on whether the option 'report sediment data as a mass fraction' in `sedgem_config.par` has been selected or not, while isotopic components are reported as per mil (). The locations (if

[20] For 'historical' reasons, the maximum number of time-series (and time-slice) data points was set to 4096. This is set by the parameter `n_data_max` in `biogem_lib.f90`, and can be altered if required.

[21] Many of the results files are created during initialization and are periodically updated (appended to) during the experiment. Even before the experiment has finished, they may contain data that is useful to view.

[22] The mid-points at which time-slices are saved are specified as described above.

[23] The mid-points at which time-slices are saved are specified as described above.

[24] This data is saved only at the termination of an experiment (i.e., the netCDF file contains only a single time-slice).

any) of sediment cores to be saved is specified in a plain text (ASCII) file pointed to by the string value of the namelist parameter `sg_par_sedcore_save_mask_name`[25]. In the mask file, a '1' indicates a location to save a sediment core at, and a '0' indicates that no sdiment core should be saved at this location. This file must be present, so to save no sediment cores, simply populate the file with all zeros in an `xx` by `yy` grid.

3. `sedcore_*` These files contain pseudo time-series of surface sediment environmental properties at each of the requested sediment core locations (if any).
4. `seddiag_misc_DATA_GLOBAL.res` – A summary of mean global sedimentation, dissolution, and preservation fluxes, and surface sediment composition.
5. `seddiag_misc_DATA_FULL.res` – Surface sediment and bottom water properties at each and every sediment grid point.

5.4 netCDF format output

The primary format for saving spatial (2- and 3-D) data by BIOGEM and SEDGEM binary in netCDF (network Common Data Form). More information on the netCDF format and the libraries necessary to compile the model can be found here. The writing of netCDF follows roughly the CF1.0 convention (NetCDF Climate and Forecast (CF) Metadata Convention). The netCDF output is written for BIOGEM and SEDGEM separately and both modules have a flag that suppresses data file saving in ASCII format, with netCDF format being the default.

Under unix/linux, the contents of a netCDF file can always be seen with;

```
$ ncdump -h filename.nc
```

This will provide the header information of the file. The command is included in the netCDF library which has to be present to run the model anyway. It's useful to get the NCO software package helping to concatenate files or extract variables as shell command. A full list of available software to manipulate or graphically illustrate netCDF files can be found here.

If you really really must insist on using Windozzzz, a 'recommended' viewer for netCDF is Panoply. In fact, the can be run under linux and on the Mac (OS X) as well. There is also ncBrowse. Again, this will also run under LINUX and on the Mac (OS X)

You can also view netCDF files from MUTLAB. However, first you will need to ensure that you have the netCDF toolbox for MUTLAB. Once the toolbox is included, you can simply load netCDF files into MUTLAB with the command:

```
>> ncload('filename','parametername')
```

Or, write it in the form:

```
>> vari=ncload('filename','parametername')
```

where `vari` is going to be a cell array. Whatever. For more information on using the MUTLAB NetCDF Toolbox, also see here.

5.4.1 Plotting with Panoply

5.4.2 MUTLAB plotting

There are some plotting scripts available for BIOGEM and SEDGEM netCDF output which can be found in `~/genie-matlab`. The plotting scripts are all written as functions and have on-line 'help', i.e., type:

```
>> help functionname
```

where `functionname.m` is one of:

[25] The location of this file is specified by the SEDGEM data input directory namelist parameter: `sg_par_indir_name` which by default is `~/genie-sedgem/data/input`.

- `plot_fields_biogem_2d.m` – Plot a 2-D field from `fields_biogem_2d.nc`
- `plot_fields_biogem_3d_i.m` – Plot a vertical-meridional (2-D) slice through the ocean (i.e., all cells have the same `i` (longitudinal) coordinate value) from `fields_biogem_2d.nc`.
- `plot_fields_biogem_3d_i_AVE.m` – This is based on `plot_fields_biogem_3d_i.m` but plots a slice constructed by averaging longitudinally over a supplied mask.
  For example: if the mask is of the entire ocean (`mask_worbe2_ALL.dat`), the result is a global meridional cross-sectional mean. If the mask is just of a single basin such as the Atlantic (`mask_worbe2_Atlantic.dat`), the result is the Atlantic meridional cross-sectional mean. Masks can also be constructed that are only a single cell 'wide' longtudinally, but which take a meandering path following an ocean transect[26]. The trivial usage would be to construct a mask consisting of a vertical slice with equal `i` – the result is then in effect exactly the same as using `plot_fields_biogem_3d_i.m`.
  The mask is a 2-D array of model grid points (on the BIOGEM grid). A value of '1' represents a vertical column of ocean cells to include, whereas a value '0' will exclude all cells in the water column at that particular grid point.
- `plot_fields_biogem_3d_k.m` – Plot a horizontal slice through the ocean (i.e., all cells having the same `k` (vertical) coordinate) from `fields_biogem_2d.nc`. Note that for the 8-layer version of the ocean circulation model, legal values for `k` range from 1 through 8, with `k=8` representing the uppermost (surface layer). For the 16-level version of GOLDSTEIN, legal values for `k` are 1-16.
- `plot_fields_sedgem_2d.m` – plot a 2-D fields from fields_sedgem_2d.nc

5.5 Re-start files

Re-start files are saved in the results directories by default. For ATCHEM, BIOGEM, and SEDGEM, these are in a binary format (`form='unformatted'`), and the filenames take no extension (and are simply named after the module). For the climate modules of GENIE-1 (GOLDSTEIN, SEAICE, EMBM), re-start files can be saved in either plain text (ASCII) or netCDF format.

[26]e.g., as in file: `mask_worbe2_WOCE_P16.dat`

A Quick-start guide

The quick-start guide assumes that you already have access to the SVN code repository, i.e., have a username and password assigned. If not – get one! You will not get very far without any code ...

1. Log into your computing account
2. Check out the 'trunk' source code for GENIE From your home directory (~):
   ```
   $ svn co http://source.ggy.bris.ac.uk/subversion/genie/trunk
   --username=YOURNAME genie
   ```
   (all one line) where `YOURNAME` will be your username for the SVN code repository. You will be asked for a password.
3. Change directory to `~/genie/genie-main` and:
   ```
   $ make
   ```
   This compiles the default configuration of GENIE. It serves to check that you have the software environment correctly configured. If you are unsuccessful here ... too bad. Try editing:
   `user.mak`
   `user.sh`
   which are located in `~/genie/genie-main` and which set the environment.
4. Next:
   ```
   $ make assumedgood
   ```
   This creates a 'gold standard' set of experimental results for GENIE-2, against which your version of GENIE-2 can subsequently be check to ensure that you have not broken it in any way. These results cannot be stored centrally (on SVN) because GENIE-2, incorporating a dynamical atmosphere, is chaotic. Now:
   ```
   $ make test
   ```
   These are short tests that check against your newly created 'assumed good' experimental results. You should get a pass here. Followed by:
   ```
   $ make testbiogem
   ```
   This is a short experiment using GENIE-1 and incorporating BIOGEM. Alternatively, `make testebgogs` will run a (EMBM-based) climate only test. Because the results are not chaotic, the test compares the result against a 'known good' set of results files that are held on SVN and installed along with the code in `~/genie/genie-knowngood`.
5. At this point, the science modules are currently compiled in a grid and/or number of tracers configuration that is unlikely to be what you want for running experiments. Clean up all the compiled GENIE modules, ready for re-compiling from the source code, by:
   ```
   $ make cleanall
   ```
6. Change back to your home directory (`cd` or `cd~`).
7. Downloaded the `old_rungenie.sh` script from `mygenie.seao2.org` (under 'GENIE resources: MODEL CONFIGURATIONS AND EXPERIMENTAL DESIGNS'). This file **MUST** be executable:
   ```
   $ chmod u+x old_rungenie.sh
   ```
8. For running GENIE-1 as described in this manual[27], you will need to create the following directories:
   ```
   ~/genie_archive
   ~/genie_forcings
   ~/genie_log
   ~/genie_userconfigs
   ```

[27] These directories are not an 'official' part of the GENIE distribution nor way of running the model.

You will notice that `~/genie_output` is created automatically when running GENIE for the first time.

9. Downloaded user config file (`worbe2_preindustrial` from `mygenie.seao2.org` (under 'User config (experimental design) namelist parameter files') – either just this single file, or download and unpack the tar-ball of various user config files. The user config files live in `~/genie_userconfigs`.
10. Downloaded the forcing subdirectory (`worbe2_preindustrial` from (`mygenie.seao2.org` (under ' Tracer forcing definitions') – either just this single directory AND all its contents, or download and unpack the tar-ball of various forcing subdirectories. The forcing subdirectories live in `~/genie_forcings`.
11. Finally, to run something:

```
$ ./old_rungenie.sh genie_eb_go_gs_ac_bg
~/genie_userconfigs worbe2_preindustrial 11
```

(all one line)

B Available GENIE modules

Component	Abrv.	?	Model	Reference
Atmosphere	ig eb fa		IGCM (3-D AGCM) EMBM (2-D) Fixed	*de Forster et al.* [2000] *Fanning and Weaver* (1996) —
Ocean	go so fo		GOLDSTEIN Slab Fixed	*Edwards and Marsh* (2005) *de Forster et al.* [2000] —
Sea-ice	gs ss fs		GOLDSTEIN (multi-option) Slab Fixed	*Edwards and Marsh* (2005) *de Forster et al.* [2000] —
Land surface	— ml el fl		GENIE-land IGCM-land ENTS Fixed	*Meissner et al.* [2003] *de Forster et al.* [2000] *Williamson et al.* [2006] —
Ice-sheets	gi fi		GLIMMER Fixed	*Payne* (1999)
Vegetation	— — — —		— TRIFFID ENTS Fixed	— *Cox* [1998] *Williamson et al.* [2006] —
Ocean biogeochemistry	bg		BIOGEM	*Ridgwell et al.* [2007a]
Deep-sea sediments	sg		SEDGEM	*Ridgwell and Hargreaves* [2007]
Atmospheric chemistry	ac		ATCHEM	*Ridgwell et al.* [2007a]
Terrestrial weathering	rg		ROKGEM	—

Figure 2: Table summarizing the different GENIE modules.

C GENIE directory structure

This section describes the directories and files pertinent to configuring and running biogeochemistry experiments in GENIE-1.

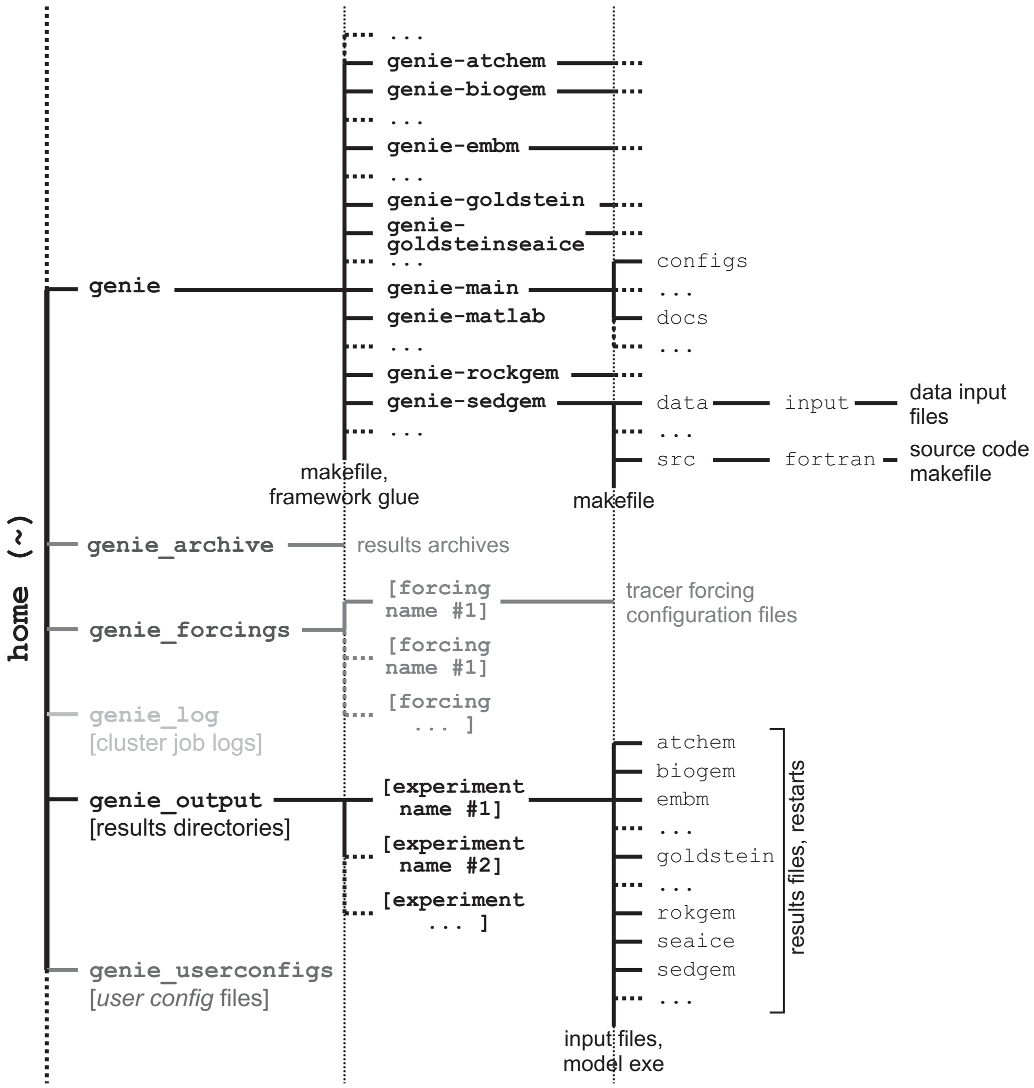

Figure 3: Schematic of the GENIE file-tree. Note that only the sub-directory structure for the `genie-biogem` module is shown expanded; near-identical sub-structures are present in all the science modules. The dark gray shaded parts of the directory tree are not part of the official GENIE release, but instead required when using the `old_rungenie.sh` script. The light gray branch is required only when submitting cluster jobs exactly as described in this document. The directory in which the genie code tree is installed is assumed to be your home directory and is designated by ˜.

The details of the directory contents are as follows.

- `~/genie/genie-atchem/data/input`
  Input data files for the atmospheric chemistry module ATCHEM. Currently there are no data files that are required to be present here.
- `~/genie/genie-atchem/src/fortran`
  Source code for the atmospheric chemistry module ATCHEM:
  - `atchem.f90` – Updating of atmospheric composition; re-start file saving.

- atchem_box.f90 – [currently just an empty template module - will hold atmospheric 'chemistry' subroutines]
- atchem_data.f90 – Load re-start; initialize atmospheric grid and tracers; read in namelist settings.
- atchem_lib.f90 – Parameter and array definitions having ATCHEM module scope; set namelist definitions.
- cpl_comp_atmocn.f90 – Set surface-atmosphere trace-gas interface composition.
- cpl_flux_ocnatm.f90 – Integrate net trace-gas flux to atmosphere from surface.
- end_atchem.f90 – [currently does not 'do' anything except print-to-screen]
- initialise_atchem.f90 – Initialize ATCHEM; read in name-list settings.
- makefile – Errr, the makefile ... :o)

• ~/genie/genie-biogem/data/input
Input data files for the ocean biogeochemistry module BIOGEM. This is the location searched for input data files if the default input directory namelist setting is used, which is set by the namelist parameter: bg_par_indir_name. In addition to any files added locally (e.g., for specific model runs) the SVN server holds the following files:

- Default[28] configuration of tracer forcing of BIOGEM:
  * configure_forcings_atm.dat – Selection and configuration of forcing of atmospheric (gas) tracers.
  * configure_forcings_ocn.dat – Selection and configuration of forcing of ocean (dissolved) tracers.
  * configure_forcings_sed.dat – Selection and configuration of forcing of sediment (particulate matter) tracers.
- Configuration files for use in the BIOGEM test ($ make testbiogem):
  * genie_eb_go_gs_ac_bg_test_save_sig.dat – Specification of the times of time-series data saving.
  * genie_eb_go_gs_ac_bg_test_save_timeslice.dat – Specification of the times of time-slice data saving.
  * genie_eb_go_gs_ac_bg_test_windspeed.dat – Prescribed wind field for calculating air-sea gas exchange.
- Various illustrative (template) time-series data saving specification files:
  * save_sig_historical.dat – Time-series saves specified to cover 1765.5 to 2000.5 at 1.0 year intervals, with 0.5 to 1000.5 following save_sig_log10.dat (below).
  * save_sig_log10.dat – Time-series saves specified at intervals of 1.0 years up until 10.5 years, then 10.0 years until 100.5 years, etc etc.
  * save_sig_log10_25kend.dat – As above, but terminating at 25000.5 (i.e., the mid-point of the year 25000).
  * save_sig_log10_50kend.dat – As above, but terminating at 50000.5 (i.e., the mid-point of the year 25000).
- Various illustrative (template) time-slice data saving specification files:
  * save_timeslice_historical.dat – Time-slice saves specified with mid-points at: 1765.5, 1994.5, 2000.5.
  * save_timeslice_log10.dat – Time-slice saves specified with mid-points at: 0.5, 10.5, 100.5, 1000.5, etc etc.

[28]These are configured for no selected forcings.

    * save_timeslice_log10_25kend.dat – As above, but terminating at 25000.5.
    * save_timeslice_log10_50kend.dat – As above, but terminating at 50000.5.
  - Modern observed annual mean wind-speed for use as an air-sea gas exchange boundary condition in BIOGEM[29]:
    * windspeed.dat
- ~/genie/genie-biogem/src/fortran
  Source code for the ocean biogeochemistry module BIOGEM:
  - biogem.f90 - Primary ocean biogeochemical cycling subroutine/loop; saving of time-series and time-slice data; re-start file saving.
  - biogem_box.f90 – Ocean biogeochemical cycling calculations; time-dependent forcing updating; tracer 'auditing' calculations.
  - biogem_data.f90 – Load re-start; initialize ocean grid and tracers; ASCII time-series data saving; ASCII time-slice data saving; read in name-list[30].
  - biogem_data_netCDF.f90 – netCDF format time-slice data saving.
  - biogem_lib.f90 – Parameter and array definitions having biogem module scope; set namelist definitions
  - end_biogem.f90 – Final auditing and reporting.
  - initialise_biogem.f90 – Initialize BIOGEM; read in name-list settings.
  - makefile – Errr, the makefile ... :o)
- ~/genie/genie-main [see: GENIE wiki pages]
- ~/genie/genie-main/configs [see: ...]
- ~/genie/genie-main/data/input
  Input data files required by the cmngem module defining the tracers in the atmosphere, ocean, and sediments and the relationships between the different tracers:
  - tracer_define.atm
  - tracer_define.ocn
  - tracer_define.sed

  The contents of these files must not be altered (nor the files moved, unless there is a very good reason to do so ...).
  Since there never will be a 'very good reason' ... **HANDS OFF**.
- ~/genie/genie-main/src/fortran/cmngem
  Source code for the geochemistry library modules – common routines, constants, and parameters that need to be accessed by all biogeochemistry modules (i.e., ATCHEM, BIOGEM, SEDGEM, and ROKGEM). The source code files are:
  - gem_carbchem.f90 – Definition and solution of aqueous carbonate chemistry; routines for solving isotopic fractionation; routines for estimating DIC and ALK given other parameters.
  - gem_cmn.f90 – Parameter and constant definitions.
  - gem_data.f90 – Read in namelist parameters.
  - gem_netcdf.f90 – Low-level netCDF file I/O routines.
  - gem_util.f90 – Miscellaneous routines, such as for reading and writing ASCII file formats, converting between isotopic notations, numerical solution and multi-dimensional linear interpolation, initialization of tracers and inter-tracer relationships

[29] The alternative is to estimate wind-speed and hence calculate air-sea gas exchange from the wind-stress field applied to the GOLDSTEIN ocean circulation model. The wind-stress based reconstruction can be chosen by setting: bg_ctrl_force_windspeed=.false.
[30] Would more logically live in initialise_atchem.f90 ...

    - initialise_gem.f90 – Tracer property and relationship definition.

- `~/genie/genie-matlab`
  MUTLAB scripts for plotting fields from the 2-D and 3-D BIOGEM and SEDGEM netCDF results files, plus related plotting configuration files.
- `~/genie/genie-rokgem/data/input`
  ...
- `~/genie/genie-rokgem/src/fortran`
  ...
- `~/genie/genie-sedgem/data/input`
  Input data files for the (deep-sea) sediment biogeochemistry module SEDGEM. This is the location searched for input data files if the default input directory namelist setting is used, whihc is set by the namelist parameter: `sg_par_indir_name`. In addition to any files added locally (e.g., for specific model runs) the SVN server holds the following files:
    - Masks which specify the sediment grid locations at which sediment cores are generated:
        * `sedgem_save_mask.36x36` – ... .
        * `sedgem_save_mask.72x72` – ... .
    - Bioturbational mixing profile:
        * `sedgem_sed_mix_k.dat` – ... .
    - Sediment grid topography (bathymetry):
        * `sedgem_topo_D.36x36` – ... .
        * `sedgem_topo_D.72x72` – ... .
- `~/genie/genie-sedgem/src/fortran`
  Source code for the ocean sediments biogeochemistry module SEDGEM:
    - `cpl_comp_sedocn.f90` – ... .
    - `cpl_flux_sedocn.f90` – ... .
    - `end_sedgem.f90` – Final auditing and reporting.
    - `initialise_sedgem.f90` – Initialize SEDGEM; read in name-list settings.
    - `sedgem.f90` – Primary ocean biogeochemical cycling subroutine/loop; saving of time-series and time-slice data; re-start file saving.
    - `sedgem_box.f90` – Ocean biogeochemical cycling calculations; time-dependent forcing updating; tracer 'auditing' calculations.
    - `sedgem_box_archer1991_sedflx.f90` – ... .
    - `sedgem_box_ridgwell2001_sedflx.f90` – ... .
    - `sedgem_data.f90` – Load re-start; initialize ocean grid and tracers; ASCII time-series data saving; ASCII time-slice data saving; read in name-list.
    - `sedgem_data_netCDF.f90` – netCDF format time-slice data saving.
    - `sedgem_lib.f90` – Parameter and array definitions having SEDGEM module scope; set namelist definitions.
    - `sedgem_nnutils.f90` – Neural network utilities.
    - `makefile` – Errr, the makefile ... :o)
- `~/genie_archive` – This directory `IS NOT` created automatically for you and is where the `old_rungenie.sh` run script will attempt to put archived results.
- `~/genie_forcings` – This directory `IS NOT` created automatically for you.
- `~/genie_log` – The location where standard output and error streams are directed into a file. This directory `IS NOT` created automatically for you.

- `~/genie_output` – The default directory where the results will be sent. This directory `IS` created automatically for you.
- `~/genie_userconfigs`

D Namelist parameter definitions and defaults

Parameters in the model are controlled via *namelists* – lists of parameter names whose values are passed into GENIE when it initializes. The default namelist parameter values are listed in a set of Tables, which can be downloaded from *mygenie.seao2.org*. These are the values and settings that GENIE will use if not told otherwise.

To effect a change in parameter value, the parameter name is simply assigned the new value, either in the *base config* used, i.e. one of the `.config` files in:

```
~/genie/genie-main/configs
```

Or, more typically, when using the `old_rungenie.sh` model job configuration and submission script: by appending the new namelist assignment(s) to the *user config* file defining the experiment in:

```
~/genie_userconfigs
```

Either way, the (re-)assignment generally takes the form:

```
namelist = value
```

Where the value is a string, the syntax is:

```
namelist = 'string'
```

The syntax for logical flag (e.g., 'true') assignments is:

```
namelist = .true.
```

For selecting (or de-selecting) tracers, the syntax is:

```
gm_atm_select_nn = .true.
```

for an atmospheric tracer (gas). `nn` is the integer index of the tracer as detailed in the Tables. For ocean and sediment (particulate) tracers, the namelist names take the same form except with `ocn` or `sed` in the namelist parameter name.

If the number of selected tracers in the ocean is changed, so to must the value of `GOLDSTEINNTRACS`, which sets the ocean array dimension and the number of tracers that must be advected, convected, and diffused. The value of `GOLDSTEINNTRACS` must be equal to the number of selected ocean tracers (i.e., the number of `gm_ocn_select_nn` that are `.true.`) *including* temperature (T) and salinity (S). The default value is 2 (just T and S, giving climate-only simulation capabilities), but this value is adjusted in the `.config` files. Any further deviation from the ocean tracer selection requires a new value to be assigned in the *user config*.

For example, for 16 selected ocean tracers (including temperature and salinity), add the line:

```
GOLDSTEINNTRACSOPTS = '$(DEFINE)GOLDSTEINNTRACS=16'
```

to the end of the *user config* file

For setting initial values of tracers, the syntax is:

```
gm_atm_init_nn = 278.0E-6
```

in the case of an atmospheric (gaseous) tracer. Again, `nn` is the index of the tracer. Ocean tracers are initialized similarly[31]. The units of the tracer initialization parameters are given in the summary Tables.

[31] There is no user-configurable initialization of deep-sea sediment composition (in SEDGEM).

E The old_rungenie.sh script

E.1 Overview

The overall strategy for the old_rungenie.sh methodology of running GENIE is as follows:

1. Copy a .config file[32] defining the required flavor of GENIE plus configuration and calibration of the climatology from ~/genie/genie-main/configs.
2. Append a series of *namelist* parameter values changes specific to a particular model experiment, particularly those involving the biogeochemistry and carbon cycling such as the specification of a CO2 emissions forcing [33] or selection of feedbacks (e.g. CO2-and-climate, CO2-and-calcification). These namelist parameter alternations (compared to the defaults) are provided in the form of a 'patch' file.
   The altered configuration details are added (patched) to create a fine-tuned configuration *.config file with a name unique to the model experiment.
   For example[34], taking a basic science module flavor and configuration:
   genie_eb_go_gs_ac_bg.config
   and modifying it according to the biogeochem experiment defined by:
   ~/genie_userconfigs/worbe2_preindustrial_1
   results in the creation of a new configuration file:
   genie_eb_go_gs_ac_bg.worbe2_preindustrial_1.config
   The new configuration file will contain additional settings such as:
   - The number of time-steps in the model in order to exactly achieve your requested run length
   - Whether a re-start file is to be used or not, and if so, where the re-start files are located.
   - The location of parameter files required by the various science modules.
3. Finally, the working directory is changed to:
   ~/genie/genie-main
   and the model is automatically invoked in the 'usual way', i.e.:
   ```
   $ ./genie_example.job -f
   configs/genie_eb_go_gs_ac_bg.worbe2_preindustrial_1.config
   ```
   BUT NOTE that this is to illustrate what the rungenie.sh script does automatically for you and DOES NOT mean that you should necessarily enter in the ./genie_example.job -f command by hand (unless you have completely configured an experiment manually).

E.2 technical details

In glorious and wonderful and completely tedious detail, old_rungenie.sh actually does the following:

1. ACCEPT PASSED PARAMETERS
   Check that the required parameters (4) have been passed. Set local variables derived from passed parameters.
2. SET LOCAL FILE AND DIRECTORY NAMES
3. CREATE EXPERIMENT CONFIGURATION FILE
   Make a copy of the specified template configuration file (passed parameter #1). Give the configu-

[32]As outlined earlier, the specification of particular flavors (unique combinations of science modules) are contained in .config files with names of the form genie_aa_oo_ss_xxxx.config. For instance, a configuration + parameter calibration of the 16-level version of the GOLDSTEIN ocean model together with ocean biogeochemistry, is defined in genie_eb_go_gs_ac_bg_itfclsd_16l_JH.config

[33]The details of any biogeochemical forcings are stored in a directory specified by the value of the namelist parameter bg_par_fordir_name.

[34]Refer to the tutorial for a detailed description of this particular model experiment.

ration file a unique filename by appending the model run ID (passed parameter #3).

4. SET MODEL TIME-STEPPING
   Set up the time-stepping control of model output, data saving, and experiment run length:

   - (a) Set run length in BIOGEM:
     `bg_par_misc_t_start` is the start year, which is assumed to be zero by default.
     `bg_par_misc_t_runtime` is the run length (years) – its value is taken from passed parameter #4.
   - (b) Based on the run length, a consistent overall GENIE run length in terms of the number of internal time-steps is set:
     `ma_koverall_total`, the overall number of internal time-steps in the model, is calculated based on the run length and length of each internal time-step.
     The length of each time-step is determined by the value of `ma_genie_timestep` (in seconds), and is defined in the `.config` file by:
     `ma_genie_timestep = 365.25*24.0/500 * 3600.0`,
     i.e.,
     `ma_genie_timestep=63115.2`
     giving 500 time-steps per year[35].
     Note that also in `.config` file: `ma_ksic_loop=5` sets the updating of sea-ice to occur every 5 GENIE time-steps, and `ma_kocn_loop=5` set the updating of ocean circulation to occur every 5 GENIE time-steps.
     The `runtime_defaults.sh` value of `ma_katm_loop=1` is unchanged, meaning that the atmosphere (the EMBM in this case) is updated each and every GENIE time-step.
     Finally, `ma_dt_write` sets a default interval of output (in multiples of the GENIE time-step), and is set equal to the total number of time-steps (`ma_genie_timestep`) to give output writing at the end of the experiment only.
   - (c) Set the climate model components' restart saving frequency (iwstp):
     `ea_4`, `go_4`, and `gs_4` set the frequency of restart saving (multiples of the GENIE time-step). These are set equal to `ma_genie_timestep` to give re-start saving only at the end of the run.
   - (d) Set climate model components 'health check' frequency (npstp):
     `ea_3`, `go_3`, and `gs_3` set the frequency of 'health check' diagnostics reporting.
     These are set equal to `ma_genie_timestep+1`, effectively disabling this feature.
   - (e) Set climate model components 'time series' frequency (itstp):
     `ea_5`, `go_5`, and `gs_5` set the frequeny of 'integral diagnostics' reporting.
     These are set equal to `ma_genie_timestep+1`, effectively disabling this feature.
   - (f) Set climate model components 'average' frequency (ianav):
     `ea_6`, `go_6`, and `gs_6` set the 'output averaging' interval.
     These are set equal to `ma_genie_timestep+1`, effectively disabling this feature.
   - (g) Set ROKGEM terrestrial weathering model component reporting frequency:
     The value of `rg_par_screen_output` is set equal to `ma_genie_timestep+1`, effectively disabling this feature.

5. SET CLIMATE MODEL RE-START FILE DETAILS

   - (a) Set netCDF restart saving flag:
     The `'y'/'n'` value of `ea_31`, `go_19`, and `gs_14` determine whether a netCDF format restart file is to be saved. These are set to `'n'`.
   - (b) Set ASCII restart output flag:
     The `'y'/'n'` value of `ea_32`, `go_20`, and `gs_15` determine whether an ASCII format restart file is to be saved.

[35] For GENIE-1 flavors; a different time-step is used for GENIE-2 flavors.

- (c) Set ASCII 'restart number':
  ea_29, go_17, and gs_12 contain strings used to form the ASCII restart file names.
  The string is appented by *.n*, where *n* is an integer. The value of *n* is incremented at each new re-start save. However, because the frequency of re-start saving has been configured to create only a single restart save at the end of the run, the extension of the restart file will always be .1 when using rungenie.sh (unmodified).

6. CONFIGURE USE OF RESTART
   Configure GENIE to use a restart (the alternative being to start from 'cold').
   Restart namelist items are set according to the presence or absence of the 5th passed parameter (restart experiment name). If the 5th parameter is present when rungenie.sh is invoked, the following action is taken:
   - (a) Check that the restart experiment (directory) exists. Generate an error message and exit if not.
   - (b) Set the climate model components to use a restart, namelist items:
     ea_7, go_7, gs_7
     The syntax is 'c' for a continuing run (i.e., using a restart), and 'n' for new (from cold).
   - (c) Set the biogeochemistry model components to use a restart, namelist items:
     ac_ctrl_continuing, bg_ctrl_continuing, sg_ctrl_continuing, (rk_ctrl_continuing)
     The syntax is LOGICAL (.true. or .false., or t or f)
   - (d) Set whether a netCDF restart is used for the climate model components:
     ea_30, go_18, gs_13
     The syntax is 'n' for no netCDF restart file format (instead it will be ASCII).
   - (e) Set the climate model ASCII restart file name namelist values:
     ea_35, go_23, gs_18.
     The name in each case is 'rst.1' (rst being the default saved restart filename, and .1 indicating the first saved restart (see above).
   - (f) Set re-start location (given by optional parameter #5).
     The location of the re-start file for each module is given by a namelist parameter[36]: xx_rstdir_name in the case of the climate model modules and where xx is the module abbreviation (ea, go, gs), and yy_par_rstdir_name in the case of the biogeochemical model modules, with yy being the module abbreviation (ac, bg, sg, rg).

   If restarts are not to be used, then:
   ea_7, go_7, gs_7 are all set to 'n', and
   ac_ctrl_continuing, bg_ctrl_continuing, sg_ctrl_continuing, rk_ctrl_continuing are all set to .false..

7. APPEND EXPERIMENT SPECIFIC NAMELIST CHANGES
   Append the contents of the user configuration file specified by the run ID (passed parameter #3) and its directory location (passed parameter #2) to the basic flavor and configuration file (parameter #1).
   But first ... in case one has a Windoz user infestation ... the namelist file is conditioned to avoid possibility of carriage return/line-feed screw-ups[37].

8. GO!
   Run the model! Change directory and from ~/genie/genie-main, invoke:
   ```
   ./genie_example.job -f
   configs/genie_eb_go_gs_ac_bg.worbe2_preindustrial_1.config
   ```

9. CLEAN UP
   Move the configuration file created to the results directory in genie_output, and archive entire

[36] A string describing the full path + filename - see page 28.
[37] This step can be omitted (commented out/deleted) in the event that no Windoz users are involved

results directory of the run, by:
`tar cfz genie_archive/$MODELID.$RUNID.tar.gz $OUTPUTPATH`

This line will need to be edited if the instaled directory structure differs from the default assumed in this manual. Or this line can simply be commented out (`#`) if nto required.

F FAQ (aka: has this dumb question been asked before?)

F.1 Running experiments: General

F.1.1 Help! My experiment has died ...

If, when using the `old_rungenie.sh` shell script to run a GENIE-1 experiment (at the command line / interactively), it all goes horribly pear-shaped ...

1. Dies absolutely immediately. Check that the `old_rungenie.sh` shell script has executable permissions. Also check that the directory you are trying to run the model from is your home directory (˜).
2. Does it die almost immediately, before the line:
`>> Here we go ...`
appears? If so, there should be an error message telling you that a particular file or directory cannot be found. Check:
    - All the files and directories you have specified exist.
    - You have not omitted spaces where you should not have, nor added spaces where a '_' separator was required.
    - You have not mis-spelt anything – a common cause of problems is in reading the number one ('`1`') for the letter el ('`l`'), or *vice versa* in the computer font (Courier).
3. Another potential source of files not being found, either as the run script starts, or as GENIE initializes, is that '`~`' may not necessarily get expanded into the path of your home directory (e.g., '`/home/mushroom`'. In this situation, '`~`' can simply be replaced with '`$HOME`'. Note that as well as making this substitution at the command line, the user config file may also contain instances of '`~`' (such as in specifying particular forcings.
4. Further trouble can occasionally arise when using Windoz and editing files (e.g., the *user config* file) and it is possible to 'corrupted' the format of the file. For what file(s) you have edited, use the command `dos2unix` to strip off Windoz formatting characters (which are invariably invisible in most editors). The syntax for this (or see the linux Man pages, or even Google it) is
`$ dos2unix FILENAME`
5. If the model starts running, but dies with a reported failure to solve the aqueous carbonate system, it may be that you need to force a re-compile. Running GENIE with array dimensions which do not match the number of tracers selected is a common cause of failures to solve the aqueous carbonate system, as often calcium ion or other tracer concentrations become 'corrupted' and get assigned nutty and all but impossible values.

F.1.2 When do I have to recompile GENIE?

You will need to recompile GENIE in the following situations:

1. You have just carried out one of the GENIE tests, e.g., MAKE TEST or MAKE TESTBIOGEM.
2. You have changed the dimension of the climate model grid (which also means an automatic change in the biogeochemistry modules), either horizontally (e.g., going from 36x36 to 64x64) or vertically (e.g., going from 8 levels in the ocean to 16).
3. You have changed the number of selected biogeochemical tracers (either gaseous, dissolved, and/or particulate).

The latter two involve a change in compiled array dimension.

The compute nodes of the cluster do not have access to the FORTRAN compiler. Hence, submitted jobs cannot recompile modules and all science modules must be already compiled when a job is

submitted.[38]

To recompile, you must first force a clean of the compiled modules. From:

```
~/genie/genie-main
```

issue:

```
$ make cleanall
```

Now you need to recompile (and re link) the science modules. To do this, first, start an interactive run of the experiment you want to conduct. This will ensure that it is correctly compiled. This also serves as a visual check that you have requested a *user config*, restart, etc that actually exists. Start the run for the length of time you intend to use when submitting the experiment as a job to the queue, but kill it (keyboard command: **Ctrl-C**) once it is compiled and you are happy that it is running OK (say, after 10 years).

At this point you can be reasonably confident that the experiment is safe to submit the job to the cluster (and all files and inputs are as they should be).

If you have multiple experiments, all with the same resolution and number of tracers, you **DO NOT** need to re-run interactively or attempt to recompile. Also, you can add 'modules' and not recompile. i.e., you can interactively run an ocean -only carbon cycle. And then submit it. And then submit an experiment using SEDGEM as well. (Because when the model is compiled, ALL sciences modules are compiled, meaning that all there is to do is just link them, which does not require the (`ifort`) FORTRAN compiler.)

F.2 Climate

F.3 Biogeochemistry

F.3.1 Can I separate out solubility (SST) related changes from stratification and circulation changes?

With BIOGEM coupled to the climate model core of GENIE-1 [39], a change in atmospheric CO2 will induce a change in SSTs, which in turn affect the carbon cycle and feedback on CO2 via changes in solubility and via changes in circulation (stratification) and thus biological productivity. There are times when is it helpful to separate out solubility related changes from circulation related changes. This equally applies to dissolved O2 and CO2. The problem is that you need a change in climate and surface temperatures in the climate model in order to induce a change in circulation.

There is a way of having an altered climate and circulation, which then affects the marine carbon cycle, yet specify the SSTs 'seen' by BIOGEM (and thus used in solubility calculations).

First of all, control the radiative forcing of climate internally in the EMBM rather than externally by the atmospheric CO2 concentration calculated by ATCHEM. 'Turn off' explicit CO2 forcing of climate by setting: `ea_36='n'`. The namelist parameter `ea_20` will then dictate the EMBM radiative forcing: a value of 1.0 (default) gives no change in radiative forcing (CO2 = 278 ppm), a value of 2.0 corresponds to the effect of doubling CO2, 4.0 x4 CO2, etc. Altering the value of `ea_20` thus lets you control climate (and circulation) without having to adjust CO2 and the carbon cycle.

Next, SSTs in BIOGEM can be specified independently of the climate model. You achieve this by setting up a restoring forcing of ocean temperatures at the surface. Note that by default, prescribing SSTs (or SSSs) in BIOGEM does not propagate through to the climate model which does its own independent climate thing based on the value of `ea_20`. This allows you to retain the surface temperatures and thus solubility associated with a x1 CO2 World, but have a warmer more stratified ocean (appropriate for a much warmer World).

[38] It is OK to change the flavor of GENIE as linking is done by the C compiler.
[39] Namelist: `ea_36='y'`

What actually happens is that BIOGEM receives both the altered circulation field and the altered SSTs due to x4CO2, but sets its own SSTs internally rather than use those calculated by the climate model. Setting up the SST restoring is in principal just like for PO4. The values for the SST field you can simply copy and paste out of a prior x1CO2 experiment.

The converse experiment, is to have circulation and biological productivity not change, but explore the effect of changes in SST-driven solubility. i.e., to separate the solubility pump from circulation change effects on glacial CO2.

F.3.2 What is 'tracer auditing' – should I have it switched on?

When developing a new model parameterization, it is of fundamental importance that careful track is kept of the total tracer inventory of the system in the face of internal mass transfer and any inputs (e.g., prescribed restoring or flux boundary conditions) or outputs (e.g., sedimentation). No spurious gain or loss of tracer mass must occur as a result of bugs introduced to the code. The tracer inventories of the ocean can be periodically calculated and compared to that predicted to have occurred on the basis of any net input (or output) occurring in the intervening time to help catch bugs. The simplest implementation would be an audit carried out at system start-up (before any transformation of tracer mass has taken place), and at the very end (after the last transformation of the tracer fields). However, integrating over over an extended time period can lead to the excessive accumulation of numerical (truncation) errors. Instead, the audits are carried out periodically during the model run. The periodicity of tracer auditing follows the times specified for time-series data saving (i.e., at time listed in the file specified by `bg_par_infile_sig_name`).

The entire audit procedure is as follows:

1. First, an initial inventory is calculated, achieved by summing the product of the concentration of each (selected) tracer with the mass of each each cell, across all wet cells.
2. During the model run, the net spatially- and time-integrated transfer of tracer mass arising from all transfers across the external reservoir boundaries is calculated.
3. At a periodic pre-defined time, the inventories are re-calculated. The difference between old and new inventories should be equal to the integrated net flux. If the relative difference between re-calculated inventory and estimated (on the basis of net flux) differs by more than a predefined threshold then an error message is raised (and the model halted if requested)
4. The integrated net flux variable is re-set to zero and steps (2-4) repeated.

In short – if you are not modifying the code then you can take it on trust(!) that the model distribution is free of (major) bugs and that spurious gain or loss of tracers does not occur. It you don't trust me ... then switch the auditing feature on.

Auditing is inactivated by default. To activate it:

```
bg_ctrl_audit = .true.
```

To adjust the threshold (relative) tolorance[40]:

```
bg_par_misc_audit_relerr = value
```

To halt the model[41] if it fails the tracer drift tolorance:

```
bg_ctrl_audit_fatal = .true.
```

A secondary benefit of tracer auditing when running the model interactively, is that it reports back to you the maximum and minimum value of all the tracers (and locations of where this occurs), as follows:

```
>>> SAVING BIOGEM TIME-SERIES @ year  0.50  278.069  -6.501  16.522  3.843  18.543  ...
    temp            / min = 0.2713E+03 (18,36, 8) / max = 0.3030E+03 ( 4,18, 8)
```

[40] By default, this is set to `1.0E-08`.

[41] By default the model will continue running, even if there is an apparent spurious drift in tracer inventories occurring.

```
sal             / min = 0.3337E+02 (10,35, 8) / max = 0.3891E+02 (30,29, 8)
DIC             / min = 0.1878E-02 (35,24, 8) / max = 0.2581E-02 (33,21, 1)
DIC_13C         / min = -.4225E+00 ( 3,16, 3) / max = 0.2792E+01 (25,13, 8)
DIC_14C         / min = -.1779E+03 (33,21, 1) / max = 0.2197E+02 (30,29, 8)
PO4             / min = 0.7071E-07 (29,28, 8) / max = 0.3806E-05 ( 3,16, 3)
O2              / min = -.4521E-04 (27,30, 5) / max = 0.3363E-03 (24,35, 8)
ALK             / min = 0.2212E-02 (10,35, 8) / max = 0.2724E-02 (33,21, 1)
DOM_C           / min = -.4159E-05 (21,34, 3) / max = 0.1517E-04 (32,25, 8)
DOM_C_13C       / min = -.1000E+20 ( 1, 3, 2) / max = 0.5817E+01 (29,36, 8)
DOM_C_14C       / min = -.1000E+20 ( 1, 3, 2) / max = 0.2236E+04 (29,36, 8)
DOM_P           / min = -.3924E-07 (21,34, 3) / max = 0.1431E-06 (32,25, 8)
Ca              / min = 0.9769E-02 (10,35, 8) / max = 0.1136E-01 (30,29, 8)
CFC11           / min = 0.0000E+00 ( 1, 3, 2) / max = 0.0000E+00 ( 1, 3, 2)
CFC12           / min = 0.0000E+00 ( 1, 3, 2) / max = 0.0000E+00 ( 1, 3, 2)
Mg              / min = 0.5050E-01 (10,35, 8) / max = 0.5888E-01 (30,29, 8)
```

F.3.3 How do I do an ocean CO2 injection experiment?

There is a hard way (but maximum flexibility), a less hard way, ... and an easy way. To cut the shit – what follows is the easy way!

First, you want to use the updated tracer forcing format:

```
bg_ctrl_force_oldformat=.false.
```

Put this line in the user config file if it is not already there, perhaps under 'FORCINGS' section.

You will need a forcing template for the CO2 injection – `pyyyyz_FCO2_UNIFORM`. This is provided on mygenie.seao2.org. Download and unpack (`tar xfz pyyyyz_FCO2_UNIFORM.tar.gz`) from the directory: `~/genie_forcings`. As it stands, this is configured to stuff 1 PgC yr-1 of CO2 into the ocean over the course of one year. The location of the CO2 injection is some random default place that probably does not exist, which is not very good. So, you need to specify your ocean location. For this, add the following lines to a *user config* file:

```
bg_par_force_point_i=22
bg_par_force_point_j=33
bg_par_force_point_k=5
```

which corresponds to a cell in the N. Atlantic (i,j, = 22,33) at an intermediate depth (k=5). The i,j,k coordinates are counted from left-to-right with longitude: i, from bottom to top with latitude: j, and form top to bottom with depth for ocean level, k. The land-sea mask and maximum depth (lowest k integer) you are allowed can be got from the BIOGEM 2D netCDF, variable `grid_level`. This is a map of the 'k' values. ¿90 means land, for the 8-level ocean the ocean depths will be between 1 and 8. 8 being the surface. So the map is of the depth of the ocean and thus lowest k value you are allowed to use.

By default, using the CO2 injection forcing template you will get 1 PgC emitted to the ocean, in the location you specify. You can scale the amount of carbon up via the namelist parameter:

```
bg_par_ocn_force_scale_val_3=xxx
```

where `xxx` is the multiple of 1 PgC you want to inject. NOT your favorite movie viewer rating. e.g., 100 PgC:

```
bg_par_ocn_force_scale_val_3=100.0
```

Note that 100.0 PgC is quite a lot of carbon to be injecting into a single location (cell) in the ocean model! By default, the time-scale of injection is set as 1 year. To increase the time over which the CO2 injection takes place use the namelist parameter `bg_par_ocn_force_scale_time_3`, which simply scales the time interval. i.e.,

```
bg_par_ocn_force_scale_time_3=10.0
```

causes the CO2 injection to take place over 10 years. But since the flux is in units of PgC per year, you will get 1000.0 PgC carbon total (10 years x 100 PgC yr-1). So a combination of both namelist scaling parameters (both flux scaling, and interval scaling) will be needed for the required total CO2 injection.

Note that the integer at the end of the namelist parameter name corresponds to the index of the ocean tracer. `3` is DIC. `12` would allow you to inject alkalinity into the ocean (but the you would need to create additional forcing specification files).

The slightly harder way involves entering in the i,j,k location explicitly in the forcing configuration file `configure_forcings_ocn.dat`. Altering the magnitude and/or duration of the flux release requires editing `biogem_force_flux_ocn_DIC_sig.dat`.

The hardest way requires that two 3D fields explicitly specifying the spatial nature of the forcing flux are created and modified.

For these alternative options – see earlier section on tracer forcings (Section 4).

F.3.4 I have been told 'ERROR MESSAGE: Particulate tracer CaCO3 does does not have the corresponding ocean tracer Ca selected' – is this a problem ... ?

No! You are simply being reminded that you have calcium carbon ($CaCO_3$) selected as a particulate tracer in the model, but although when it dissolves it releases Ca^{2+} (and removes Ca^{2+} when $CaCO_3$ is precipitated), you do not have Ca^{2+} selected as an explicit dissolved tracer in the ocean. This is not a problem as by far the most important effect on the carbon cycle of adding/subtracting Ca2+ is a change in alkalinity, which is implicitly account for. Only on **very** long time-scales, or in deep-time situations when the Ca^{2+}/Mg^{2+} ratio was very different form today, might you need to select Ca^{2+} (and Mg^{2+}) as an ocean tracer.

F.4 Running experiments: The `almond.ggy.bris.ac.uk` cluster

F.4.1 Do I have to submit experiments to the queue rather than running interactively?

Yes! Except for developing the model and debugging, testing new experimental designs, and forcing a re-compile. The number of instances of the model that can be run simultaneously interactively is limited by the number of processing cores (4) on the head node. The more experiments that are run interactively, the slower everything will go. Additionally, if you even temporarily lose your internet connection, an interactively-run experiment will die. The queue is there for your convenience, believe it or not ...

F.4.2 Can I leave all my experiment results on the cluster for ever?

NO! *Nothing* is backed up on the cluster, and space is not infinite. So, periodically, transfer archived (`.tar.gz`) results off of the cluster and delete both the archive file and the results directory.

G Known issues

G.1 Radiocarbon tolerance

The comparison tolerance for 'make testbiogem' has been raised (01/05/08) from 5 to 19 Ulps (units of last place–think of notches on a ruler marked with numbers resolved by a variable of so many bytes) so that the test will pass when using the g95 compiler (version 0.91). The variables in question is ocn_DIC_14C.

G.2 Ocean tracer number related compilation problems

If the number of dissolved ('ocean') tracers (or the grid resolution) is changed then the model will need to be recompiled from 'clean'. This is because the ocean circulation model (GOLDSTEIN) is compiled with array dimensions sufficient only for the actual number of selected tracers in the ocean (including temperature and salinity). Theoretically, it might be possible to 'clean' only the GOLDSTEIN and BIOGEM modules as well as the interfacing 'glue'. However, it is much safer to request that all modules (libraries, objects, and dependency information) is cleaned up. You can do this from the command line (within `~/genie/genie-main`) by issuing the command:

```
$ make cleanall
```

One symptom of an incorrectly-compiled number of tracers is page-after-page-after-page-after-page- ... of output looking something like:

```
*** WARNING ***
-> Originating location in code [module,subroutine]: gem_carbchem.f90,sub_calc_carb
-> ERROR MESSAGE: Numerical instability at step; 0001 / Data; dum_DIC,dum_ALK,
dum_Ca,dum_PO4tot,dum_SiO2tot,dum_Btot,dum_SO4tot,dum_Ftot,dum_H2Stot,dum_NH4to
t,loc_H4BO4,loc_OH,loc_HPO4,2.0*loc_PO4,loc_H3SiO4,loc_HN3,loc_HS,loc_H,loc_HSO
4,loc_HF,loc_H3PO4,pH(SWS), pHfree, pHtotal, pH (OLD), pH (guess #1), pH (guess
#2),[CO2],[CO32-],[HCO3-]
-> ERROR DATA:      1.801730109436933E-007
-> ERROR DATA:      1.575704792562387E-004
-> ...
```

On the other hand, *appropriate* output from BIOGEM looks like (depending on the exact options set) something rather more like:

```
>>> SAVING BIOGEM TIME-SERIES @ year 0.50  493.088  0.000  14.659  4.479  18.815  34.815  ...
>>> SAVING BIOGEM TIME-SLICE @ year  0.500000000000000
>>> SAVING BIOGEM TIME-SERIES @ year 1.50  707.058  0.000  14.659  4.553  18.831  34.815  ...
```

G.3 Stack space

You may encounter issues with regards to the `ifort` Intel FORTRAN compiler (an maybe others), particularly when using SEDGEM because of the size of the arrays holding sediment information:
"*The Intel Fortran Compilers 8.0 or higher allocate more temporaries on the stack than previous Intel Fortran compilers. Temporaries include automatic arrays and array sub-sections corresponding to actual arguments. If the program is not afforded adequate stack space at runtime relative to the total size of the temporaries, the program will terminate with a segmentation fault.*"
The (a?) solution is to increase the CPU stack space, Try:

```
$ ulimit -s unlimited
```

G.4 Re-starts

There is a minor bug associated in how re-starts are done. The shortwave radiation incidence at the ocean surface is used to calculate biological productivity. The spatial field of SW radiation is passed to BIOGEM after the biogeochemical ocean update has been carried out. Although physical boundary conditions such as SW radiation are made available to BIOGEM during initialization, the SW radiation field is only calculated during the time-stepping loop. Thus, the very first time-step that BIOGEM takes

has zero in the SW radiation field. This affects experiments from a 'cold' start, but only trivially. More noticeable is that from a re-start, the ocean carbon cycle experiences a brief 'hick-up' with no biological productivity associated with the very first time-step.

This bug was not present in older version of the model, and was introduced associated with a re-ordering of the time-stepping of BIOGEM relative to the climate model components.

H Contact Information

- Andy Ridgwell: `bandy@seao2.org`
- Gethin Williams: `gethin.williams@bristol.ac.uk`